

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ
(ініціали, прізвище)

“ ____ ” _____ 202__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системне програмування»

спеціальності

123 «Комп'ютерна інженерія»

на тему: «Модуль розпізнавання зображення УЗД»

Виконав:

студент IV курсу, групи КВ-63

Логінов Олександр Сергійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник Професор кафедри СПіСКС, д.т.н. Терейковський І.А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студента

Логінов Олександр Сергійович

(прізвище, ім'я, по батькові)

1. Тема проєкту: Модуль розпізнавання зображення УЗД

керівник проєкту: Професор кафедри СПіСКС, д.т.н., професор
Терейковський І.А.

затверджені наказом по університету від «25» Травня 2020 р. № N1181-С

2. Термін подання студентом проєкту _____

4. Зміст пояснювальної записки: огляд проблеми, постановка задачі, аналіз існуючих аналогів, вибір засобів реалізації, реалізація, висновок.

5. Перелік графічного матеріалу: архітектура розробленого програмного забезпечення, діаграма взаємозв'язків класів, блок-схема роботи форми №1, блок-схема роботи форми №2.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Видача завдання на дипломне проєктування	12.11.2019	
2	Вивчення літератури за тематикою роботи	24.11.2019	
3	Розроблення та узгодження технічного завдання	19.12.2019	
4	Розроблення структури додатку	27.01.2020	
5	Розроблення дизайну та графічних елементів	11.02.2020	
6	Програмна реалізація додатку	20.03.2020	
7	Тестування додатку	26.04.2020	
8	Підготовка матеріалів текстової частини проєкту	24.04.2020	
9	Підготовка матеріалів графічної частини проєкту	07.05.2020	
10	Оформлення технічної документації проєкту	20.05.2020	
11	Попередній захист дипломного проєкту	25.05.2020	
12	Захист дипломного проєкту	18.06.2020	

Студент

(підпис)

Олександр ЛОГІНОВ

(Ім'я та ПРИЗВИЩЕ)

Керівник проєкту

(підпис)

Ігор ТЕРЕЙКОВСЬКИЙ

(Ім'я та ПРИЗВИЩЕ)

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (53с., 22 рис., 4 додатки).

Об'єкт розробки – створення програмного продукту – модуль розпізнавання зображення УЗД, функціонал якого розпізнаватиме об'єкти на зображеннях УЗД засобами спеціально навченої нейронної мережі.

Модуль дозволяє: розпізнавати зображення УЗД за допомогою навченої моделі, навчання нової моделі за допомогою підготовленої бази даних, підготовка вхідних зображень та даних для навчання нової моделі. В процесі розробки було використано мову програмування Python, а також python-модулі. PyQT5 – для розробки графічного інтерфейсу. Detecto та Pytorch – для роботи з нейронною мережею, LabelImg – для підготовки даних.

В ході розробки:

- проведено аналіз існуючих методів розпізнавання зображення УЗД;
- сформульовані вимоги до модулю розпізнавання зображення УЗД;
- розроблено графічний інтерфейс для модулю розпізнавання зображення УЗД;
- розроблена архітектура програмного забезпечення і на базі архітектури реалізований програмний код;
- створений модуль для розпізнання зображення УЗД.

Ключові слова: УЗД, медичні зображення, згортова нейронна мережа, машинне навчання, python, PyQT5, PyTorch, розпізнавання зображень.

ANNOTATION

Qualification work contains explanatory note (53 pages, 22 pictures, 4 additions).

The object of development – creation software – module for image recognition. The functionality of the module is object recognition in images of medical ultrasound. Recognition is taken with special trained Neural network.

The module provides: to recognize ultrasound images using a trained model, teaching a new model using a prepared database, preparation of input images and data for learning a new model. The Python programming language was used in the development process, as well as python modules. PyQt5 - for developing a graphical interface. Detecto and Pytorch - for working with the neural network, LabellImg - for data preparation.

During development:

- analyzed the existing methods of ultrasound image recognition;
- formulated requirements for the ultrasound image recognition module;
- developed a graphical interface for the ultrasound image recognition module;
- software architecture was developed and software code was implemented on the basis of architecture
- created a module for ultrasound image recognition.

Keywords: ultrasound, medical images, convolutional neural network, machine learning, python, PyQt5, PyTorch, image recognition.

[illegible]

3MICT

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.	2
4. ДЖЕРЕЛА РОЗРОБКИ.	2
5. ТЕХНІЧНІ ВИМОГИ.	2
5.1. Вимоги до програмного продукту, що розробляється.	2
5.2. Вимоги до апаратного забезпечення.	3
5.3. Вимоги до програмного та апаратного забезпечення користувача. . .	3
6. ЕТАПИ РОЗРОБКИ.	4

					ІАЛЦ.045490.002 ТЗ							
Змін	Арк.	№ докум.	Підпис	Дата	Модуль розпізнавання зображення УЗД Технічне завдання				Літ.	Аркуш	Аркушів	
Розробив		Логінов О.С.									1	4
Перевірив		Терейковский І.А										
Н. контроль		Клятченко Я.М.										
Затвердив		Тарасенко В.П.							КПІ ім. Ігоря Сікорського, ФПМ КВ-63			

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Модуль розпізнавання зображення УЗД».

Галузь застосування: використання в медичних центрах для підвищення рівня медичних послуг.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення модуля розпізнавання зображення УЗД. Призначення роботи полягає у розробці кінцевого продукту та підвищення рівня медичних послуг в медичних центрах.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

1.1 Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Windows 10;
- навчання нейронної моделі по вже готовій базі
- функціонал для створення бази даних
- можливість розпізнання об'єктів на медичних зображеннях засобами уже навченої моделі

					ІАЛЦ.045490.002 ТЗ	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

- можливість аналізувати результати виконання розпізнавання засобами модуля
- функціонал використання хмарних сховищ в якості потужностей для навчання моделі
- високу стабільність виконання програмного забезпечення
- довідка для користувача
- інформування користувача про важливу інформацію спеціальними сповіщеннями

5.2 Вимоги до апаратного забезпечення

- оперативна пам'ять: 1 Гб;

5.3 Вимоги до програмного та апаратного забезпечення користувача

- операційна система Windows 10;

					ІАЛЦ.045490.002 ТЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	12.11.2019
2.	Вивчення літератури за тематикою роботи	24.11.2019
3.	Розроблення та узгодження технічного завдання	19.12.2019
4.	Розроблення структури додатку	27.01.2020
5.	Розроблення дизайну та графічних елементів	11.02.2020
6.	Програмна реалізація додатку	20.03.2020
7.	Тестування додатку	26.04.2020
8.	Підготовка матеріалів текстової частини проєкту	24.04.2020
9.	Підготовка матеріалів графічної частини проєкту	07.05.2020
10.	Оформлення технічної документації проєкту	20.05.2020

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.002 ТЗ

Арк.

4

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045490.004 ПЗ	Модуль розпізнавання зображення УЗД	53		
			Пояснювальна записка			
	A4	ІАЛЦ.045490.005 Д1	Взаємодія модулів системи.	1		
			Схема структурна			
	A4	ІАЛЦ.045490.006 Д2	Діаграма в нотації UML.	1		
			Діаграма класів			
	A4	ІАЛЦ.045490.007 Д3	Блок-схема роботи форми “Програма для розпізнавання УЗІ”	1		
	A4	ІАЛЦ.045490.008 Д4	Блок-схема роботи форми “Навчання нової моделі”	1		
		Диск CD-ROM	Текст пояснювальної записки.			
			Графічний матеріал			

					ІАЛЦ.045490.003 ТП								
Змін.	Арк.	№ докум.	Підпис	Дата									
Розробив	Логінов О.С				Модуль розпізнавання зображення УЗД				Літ.	Аркуш	Аркушів		
Перевірив	Терейковский І.А										1	2	
Консульт.													
Н. контроль	Клятченко Я.М.								КПІ ім. Ігоря Сікорського, ФПМ КВ-63				
Зав. каф.	Тарасенко В.П.												
					Відомість технічного проекту								

[illegible]

ЗМІСТ

ВСТУП	2
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	3
1.1. Особливості діагностики аномалій внутрішньоутробного розвитку.	3
1.2. Різновиди вроджених вад розвитку.	5
1.3. Візуалізація медико-біологічних даних.	8
1.4. Машинне навчання.	10
1.5. Нейронні мережі як засіб розпізнавання об'єктів на зображеннях.	12
1.6. Огляд аналогів для розроблюваної системи та постановка задачі.	14
ВИСНОВКИ ДО РОЗДІЛУ 1	17
РОЗДІЛ 2. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ	18
2.1. Вибір мови програмування.....	18
2.2. Архітектура R-CNN.	20
2.4. Засіб для створення інтерфейсу.	23
2.5. Засіб для створення анотацій LabelImage.....	24
2.6. Засіб для хмарної обробки даних Google Colab.	25
ВИСНОВКИ ДО РОЗДІЛУ 2	27
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
3.1. Розробка архітектури модуля розпізнавання зображення УЗД.....	28
3.2. Розробка користувацького інтерфейсу.	29
3.3. Засоби реалізації програмного коду.....	40
3.4. Забезпечення надійності програмного продукту.	43
ВИСНОВКИ ДО РОЗДІЛУ 3	50
ЗАГАЛЬНІ ВИСНОВКИ	51
СПИСОК ДЖЕРЕЛ	52

					ІАЛЦ.045490.004 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата	Модуль розпізнавання зображення УЗД Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив		Логінов О.С.					1	53
Перевірів		Герейковський І.А.						
Н. контроль		Клятенко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ КВ-63		
Затвердив		Гарасенко В.П.						

ВСТУП

Сучасні фахівці покращують свої робочі засоби для того щоб домогтися більшої ефективності своєї роботи.

Результати роботи медиків є важливим і це дуже відповідальна робота. Точно поставлені вчасно діагнози впливають на ключові рішення в життя людей.

Сьогодні нейронні мережі допомагають вирішити цю проблему - дозволяють лікарям помічати проблеми та протидіяти їм на ранній стадії, що допомагає пацієнтам уникати вроджених дефектів у дитини.

Використовуючи, обробляючи та аналізуючи величезний обсяг даних з надзвичайно високою швидкістю, рішення, що працюють на нейронних мережах, можуть перетворити ці дані у діючі відомості, які допомагають підвищити ефективність роботи лікаря, збільшити точність діагностики та персоналізувати лікування. Для пацієнтів - майбутніх мам у цьому випадку - ці високі технології можуть означати коротший термін обстеження та позитивні діагнози, що дозволяють прискорити якісну допомогу пацієнтам.

Дана робота спрямована на продовження даної тенденції та вирішення такої проблеми як вроджені дефекти у дітей, шляхом розробки відповідного програмного забезпечення, що дозволить ефективно розпізнавати зображення УЗД на ранніх етапах вагітності.

					ІАЛЦ.045490.004 ПЗ	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Особливості діагностики аномалій внутрішньоутробного розвитку.

Внутрішньоутробний розвиток плоду є одним із найважливіших етапів життя людини. Розвиток плоду починається з моменту утворення зиготи – тобто з моменту злиття сперматозоїду та яйцеклітини. Внутрішньоутробний розвиток триває майже 40 тижнів (близько 270 днів). Весь процес розвитку поділено на триместри: I триместр триває з 1 по 13 тиждень вагітності. II триместр триває з 14 по 28 тиждень, III триместр триває з 29 по 40 тиждень вагітності.

В кожному триместрі важливо виконати скринінг розвитку плоду. В залежності від триместру можливо виділити особливі захворювання та проблеми розвитку. Один із найбільш відомих способів – ультразвукова діагностика. [1]

Ультразвукове дослідження - це метод візуалізації на основі використання височастотних звукових хвиль для отримання поперечних зображень тіла. Ультразвуковий апарат складається з наступних компонентів: монітора, клавіатури, процесора, пристрої для зберігання даних і датчика, або перетворювача.

Датчик виконує дві функції: він випускає звукові хвилі (ехосигнали) на певній частоті і вловлює повертаються відбиті звуки на частотах, що залежать від тканин, крізь які проходять хвилі. Повертається на датчик звукова хвиля перетворюється в цифрову форму і з'являється на екрані у вигляді ехосигналів або точок. Зображення з'являються в реальному часі і можуть бути отримані в будь-якій площині зображення.

Сканування за допомогою ультразвуку - це високоінформативний, неінвазивний і безпечний спосіб отримання даних про здоров'я майбутньої

матері і плоду. Метод заснований на аналізі відмінностей відображення ультразвукових хвиль від структур неоднаковою щільності. [3]

Мета УЗ-обстеження - оцінка стану жінки і її майбутньої дитини, а також виявлення можливих відхилень від норми для прийняття необхідних заходів (УЗД на патологію плода).

Ультразвукова діагностика дозволяє підтвердити факт вагітності, отримати дані про формування плоду, будову матки і придатків, а також про стан плаценти, пуповини, навколоплідних вод. Від результатів УЗД багато в чому залежить тактика ведення вагітності та підготовки до пологів, а також вибір способу розродження. [1, 2]

Для обстеження розвитку плоду обов'язковим є дотримання графіку: скринінг першого триместру відбувається на 11-13 тижні, другого триместру на 20-24 тижні, 3 триместру на 32-34 тижні.

Додаткові (позапланові) УЗ-дослідження можуть проводитися за рекомендацією лікаря на будь-якому терміні вагітності. Повторні процедури можуть призначатися для уточнення діагнозу або, при виявленні тієї чи іншої патології вагітності, для вживання необхідних заходів щодо поліпшення стану здоров'я жінки і плоду.

Перше УЗД плоду допомагає встановити точний термін вагітності, отримати дані про розмір і кількість ембріонів, почути серцебиття плода, оцінити ступінь його розвитку, виявити можливі відхилення.

Частота серцевих скорочень, формування носової кістки, товщина комірцевої області і куприка-тім'яної розмір (КТР) - вкрай важливі параметри, за якими судять про формування плоду. Істотні відхилення від норм разом із відповідними результатами лабораторних аналізів будуть вказувати на наявність патології вагітності.

На другому УЗД при вагітності лікар-гінеколог перевіряє, чи відповідають розміри плода поточному етапу вагітності. Оцінюються довжина кінцівок дитини, розміри його голови, внутрішніх органів, обсяги грудей і

					ІАЛЦ.045490.004 ПЗ	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		

живота і т.д. Не менш важливо в цей період простежити, чи нормально розвиваються внутрішні органи, хребет, серцево-судинна система.

Також під час УЗД-скринінгу 2 триместру оцінюється стан стінок і шийки матки, придатків, розташування, товщина і структура плаценти, число судин в пуповині.

Третє УЗД плоду - ключове. На цьому етапі лікар ретельно аналізує відповідність анатомічного і функціонального стану плода нормам скринінгу для даного періоду вагітності. Надзвичайно важливим є виявлення пізніх вад розвитку.

Позапланові УЗД при вагітності проводяться, якщо:

- пацієнтка відчуває болі внизу живота;
- перенесла інфекційне захворювання;
- не відчуває ворухіння плода після 20-го тижня;
- має кров'янисті і інші виділення;
- почалася передчасна пологова діяльність;
- з'явилися сутички на терміні 36-40 тижнів;
- є інші ускладнення вагітності. [2]

1.2. Різновиди вроджених вад розвитку.

Так як причини патологій у плода можуть бути обумовлені генетикою або зовнішніми факторами, розрізняються вроджені і набуті відхилення. Перші присутні з самого моменту зачаття і діагностуються найчастіше на ранніх термінах, тоді як другі можуть з'явитися у дитини і бути виявлені лікарями на будь-якому етапі вагітності.

Патології є вродженими та набутими. Далі буде розглянено детальніше кожний різновид.

					ІАЛЦ.045490.004 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

Вроджені, генетичні патології плода в медицині називається анеуплоїдія. Це відхилення від норми хромосом дитини, яке з'являється на самих ранніх етапах його внутрішньоутробного формування.

Патології, що сталися внаслідок неправильного числом хромосом:

- синдром Дауна - проблеми з 21-ю хромосомою, ознаки - слабоумство, специфічна зовнішність, затримка росту;
- синдром Патау - порушення з 13-ю хромосомою, прояви - множинні вади розвитку, ідіотія, багатопалості, проблеми зі статевими органами, глухота; хворі діти рідко доживають до 1 року;
- синдром Едвардса - патології 18-ї хромосоми, симптоми - маленькі нижня щелепа і рот, вузькі і короткі очні щілини, деформовані вушні раковини.

Хвороби, спричинені неправильним числом статевих хромосом:

- синдром Шерешевського-Тернера - відсутність у дівчинки X-хромосоми; ознаки - низькорослість, безпліддя, статевий інфантилізм, соматичні порушення;
- трисомія по X-хромосомі проявляється незначним зниженням інтелекту, психозами та шизофренією;
- трисомія по Y-хромосомі, симптоми схожі з попередньою патологією;
- синдром Клайнфельтера вражає хлопчиків, ознаки - на тілі ослаблений ріст волосся, безпліддя, статевий інфантилізм; в більшості випадків - розумова відсталість.

Буває і так, що ембріон може бути абсолютно здоровим генетично, але набуває відхилення в процесі свого утробного розвитку під впливом різних неблагополучних факторів. Такі патології називаються набутими. Це можуть бути захворювання матері, які вона перенесла під час вагітності, погана екологічна обстановка, неправильний спосіб життя і т. д.

					ІАЛЦ.045490.004 ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

Набута патологія плода при вагітності може торкнутися найрізноманітніші органи і системи. Серед найбільш поширених можна відзначити наступні:

- деформація або відсутність (повне, часткове) внутрішніх органів (найчастіше страждає головний мозок) або частин тіла (кінцівок, наприклад);
- анатомічні дефекти лицьового скелета;
- пороки серця;
- незарощення спинномозкового каналу.

В особливу групу можна виділити також відхилення від нормального внутрішньоутробного розвитку, причини яких визначити дуже складно.

До них відносяться:

- виявляється на різних етапах вагітності патологія пуповини плода: вона може бути занадто довгою або дуже короткою, випадання її петель, вузли, аномальне прикріплення, тромбоз і кісти - все це може привести до гіпоксії і загибелі дитини;
- багатоплідність (в тому числі і сіамські близнюки);
- багато- і маловоддя;
- патології плаценти: гіперплазія (її занадто велику вагу) і гіпоплазія (якщо її маса становить менше 400 гр), інфаркт, хориоангиома, трофобластичної хвороба, плацентарна недостатність;
- неправильне передлежання плода деякі лікарі теж називають патологією.

Описані у поточному пункті патології є серйозними вадами розвитку, тому скринінг процесу внутрішньоутробного розвитку це важливий та відповідальний процес.

Засобами УЗД можливо визначити не усі можливі вади, а лише деякі. В залежності від триместру УЗД дозволяє діагностувати:

- 1) на першому триместрі:
 - пороки ЦНС (аненцефалію);

					ІАЛЦ.045490.004 ПЗ	Арк.
						7
Змін.	Арк.	№ докум.	Підпис	Дата		

- відсутність брюшинної передньої стінки (гастрошизис);
- патологію хребта у плода;
- відсутність кінцівок;
- синдром Дауна;

2) на другому триместрі практично всі видимі патології плода на УЗД можуть бути діагностовані. Це пояснюється тим, що більшість внутрішніх органів і систем плоду вже добре сформовані;

3) на третьому триместрі ультразвукове дослідження може лише підтвердити або спростувати дані, отримані іншими методами (за допомогою аналізу крові, кордоцентеза, хоріонбіопсії). [4]

1.3. Візуалізація медико-біологічних даних.

Ультразвукова діагностика (УЗД) відноситься до розділу інформатики «Медична візуалізація» та є методом отримання медичних зображень.

Незалежно від способу його отримання різні медичні зображення можна розділити на одну з двох категорій: аналогові зображення та матричні зображення..

До аналогових зображень відносяться ті, які несуть у собі інформацію безперервного характеру. Це зображення на звичайних рентгенограмах, сцинтиграмах, термограмах. Аналогові сигнали - це безперервні сигнали, у них присутнє багато зайвої інформації.

Матричне зображення - це зображення, отримане за допомогою комп'ютера. Вони засновані на матриці, що зберігається в пам'яті ПК. Матричне зображення - це зображення, отримане за допомогою комп'ютерної томографії, цифрової рентгенографії, МР томографії, комп'ютерної сцинтиляційної томографії за допомогою комп'ютерної технології обробки

					ІАЛЦ.045490.004 ПЗ	Арк.
						8
Змін.	Арк.	№ докум.	Підпис	Дата		

інформації та ультразвукового сканування. Тому порівняно з моделюванням матричне зображення дискретно. Оскільки матричні зображення базуються на комп'ютерній технології, їх можна використовувати для різних комп'ютерних процесів.

Слід зазначити, що аналогове зображення може бути перетворене в матричне, і, навпаки, воно також може бути перетворене в матрицю.

Для цього використовується спеціальне обладнання: аналого-цифровий та аналого-цифровий перетворювачі. Матричне зображення формується шляхом сканування лінії електронного пучка за рядком. Це створює можливості для сприйняття зображення в реальному часі. Для цього використовується спеціальний дисплейний процесор, який підключений до головного комп'ютера через систему зв'язку (інтерфейс). Пам'ять процесора відображення організована у вигляді матриці, і кожен елемент матриці відповідає певній області її відображення. Ця основна одиниця матричного зображення, що відповідає області номера пам'яті, називається "піксель" (від англ. Pixel pixel-pixel). Тому вся область екрану дисплея є матрицею - групою пікселів. У рентгенологічній діагностиці область відображення може формуватися у такій матричній формі: 32'32, 64'64, 128'128, 256'256, 512'512, 1024'1024, 1024'1280 пікселів. Чим більша кількість пікселів в області відображення, тим вище роздільна здатність системи відображення. Кожен піксель зображення записується в пам'ять процесора дисплея з різною кількістю біт (від 2 до 16). Його візуальні характеристики та більше інформації, яку він містить про досліджуваний об'єкт. Тому 8-бітні пікселі (система байтових запису пікселів), які найчастіше використовуються в ультразвуковій діагностиці, містять 64 рівні сірого (від чорного до білого). У діагностиці радіонуклідів в основному використовують 8-бітні пікселі, що мають 256 рівнів сірого, тобто рівень сірого. Неважко підрахувати, що матричне зображення 64 x 64 пікселів при радіонуклідної діагностиці вимагає 4096 байт пам'яті та 128 x 128 пікселів - 16384 байти зображення. [5]

					ІАЛЦ.045490.004 ПЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

1.4. Машинне навчання.

Машинне навчання - це галузь штучного інтелекту в інформатиці, яка часто використовує статистичні прийоми, щоб комп'ютери могли «вчитися» з даних (тобто поступово покращувати виконання даного завдання) без явного програмування. [6]

У галузі аналізу даних машинне навчання - це метод винайдення складних моделей та алгоритмів прогнозування - у комерційних програмах він називається прогнозним аналізом. Ці аналітичні моделі дозволяють дослідникам, науковцям даних, інженерам та аналітикам "витягувати з надійних та повторюваних рішень та результатів" та виявляти "приховані уявлення", вивчаючи історичні зв'язки та тенденції даних .

Завдання машинного навчання зазвичай поділяються на дві категорії, залежно від того, чи є система навчання, навчальні "сигнали" чи "зворотній зв'язок":

1) Навчання з учителем: Щоб показати на комп'ютері приклади вводу, що дається "вчителем", та очікуваного результату, мета - навчити загальним правилам, що відображають вхід до результату. У деяких випадках вхідний сигнал може бути лише частково доступним або обмежений спеціальним зворотним зв'язком:

- Напіваавтоматичне навчання: комп'ютерам надаються лише неповні сигнали навчання: відсутність деяких (як правило, багатьох) навчальних комплектів для цільового виведення.
- Активне навчання: комп'ютер може отримувати навчальні теги лише для обмеженої кількості примірників (залежно від бюджету), і він повинен оптимізувати свій вибір об'єктів для отримання тегів. Для інтерактивного використання вони можуть надаватися для розмітки користувачам.

- Навчання з підкріпленням: дані тренувань (у вигляді винагород та штрафних санкцій) використовуються лише як зворотній зв'язок щодо поведінки програми в динамічному середовищі, наприклад, за керуванням автомобілем або грою в ігри з супротивниками.

2) Навчання без учителя: алгоритм навчання не позначений, а структуру можна знайти лише на вході. Навчання без вчителя - це самоціль (виявлення прихованих зразків у даних) або засіб кінця (навчання оцінок).

Алгоритм навчання штучної нейронної мережі, широко відомий як нейронна мережа - алгоритм навчання, натхненний віддаленими біологічними нейронними мережами. Розрахунок базується на взаємопов'язаних групах штучних нейронів, які використовують коннекціоністські методи для обробки інформації. Сучасні нейронні мережі - це нелінійні засоби статистичного моделювання даних. Вони часто використовуються для моделювання складних взаємозв'язків між входами та виходами для пошуку шаблонів даних або для виявлення статистичних структур у загальних розподілах ймовірностей невідомої кількості спостережуваних ймовірностей. [7]

Основне призначення досліджуваної системи - узагальнення з власного досвіду. У цьому випадку узагальнення - це здатність навчальної машини точно обробляти нові, невидимі приклади / завдання після набуття досвіду в наборі даних про навчання. Приклади навчання виходять з деяких зазвичай невідомих розподілів ймовірностей (вважається репрезентативним для простору випадку), і система навчання повинна побудувати загальну модель простору, щоб вона могла робити досить точні прогнози в нових випадках.

Обчислювальний аналіз алгоритмів машинного навчання та їх виконання - це галузь теоретичної інформатики, що називається теорією обчислювального навчання. Оскільки навчальний набір обмежений, а майбутнє невизначене, теорія навчання зазвичай не може гарантувати ефективність алгоритму. Навпаки, ймовірнісні рамки ефективності дуже

поширені. Одним із способів кількісної оцінки помилок узагальнення є торгівля зміщенням та дисперсією.

Для отримання найкращої ефективності при узагальненні складність гіпотези повинна відповідати складності функції бази даних. Якщо передбачувана складність нижча за цю функцію, модель ще не пристосувалася до даних. Якщо складність моделі відповідей буде збільшена, помилка навчання буде зменшена. Однак якщо припущення занадто складне, модель може бути перекваліфікована, і узагальнення погіршиться.

Окрім системи продуктивності, теоретики обчислювального навчання досліджували також складність часу та доцільність навчання. У теорії обчислювального навчання, якщо обчислення можна проводити в поліноміальний час, обчислення вважається можливим. Є два результати складності в часі. Емпіричні результати показують, що клас функцій можна дізнатись у многочлен. Негативний результат вказує на те, що певні класи не можна викладати в поліноміальний час. [6, 7]

1.5. Нейронні мережі як засіб розпізнавання об'єктів на зображеннях.

Теорія розпізнавання зображень є галуззю кібернетики, вона розробляє теоретичну основу та методи класифікації та ідентифікації об'єктів, явищ, процесів, сигналів, ситуацій тощо. Ці об'єкти характеризуються обмеженими певними характеристиками та особливостями.

Для оптичного розпізнавання зображень може використовуватися метод пошуку зовнішнього вигляду об'єкта під різними кутами, масштабами, зміщеннями тощо. Для літер потрібно сортувати шрифти, атрибути шрифту тощо.

Другий метод - це знайти контур об'єкта та вивчити його властивості (зв'язність, наявність кутів тощо)

Інший метод - використання штучних нейронних мереж. Цей метод вимагає великої кількості прикладів ідентифікаційних проблем (з правильними відповідями) або спеціальної структури нейронної мережі, яка потребує врахування деталей проблеми.

Для розпізнавання графічних зображень пристосований окремий клас нейронних мереж – згорткові нейронні мережі (ЗНМ).

Згорткова нейронна мережа використовує різноманітні багат шарові перцептрони, які розроблені так, щоб вимагати використання найменшої кількості попередньої обробки.

ЗНМ складається з вхідних і вихідних шарів та декількох прихованих шарів. Прихований шар ЗНМ зазвичай складається з шару згортки, шару агрегації, повністю пов'язаного шару і шару нормалізації.

Цей процес описується як послідовне згортання в нейронних мережах. З математичної точки зору це перехресна кореляція, а не згортання. Це важливо лише для індексу в матриці, тому який індекс має яку вагу.

Шар згортки застосовує операцію згортання до вхідного і передає результат наступному шару. Конволюція імітує реакцію одного нейрона на зорові подразники.

Кожен згортковий нейрон обробляє дані лише для свого поля отримання.

Хоча повністю пов'язані нейронні мережі прямого поширення можуть використовуватися для вивчення функцій та класифікації даних, застосувати цю архітектуру до зображень недоцільно. Через дуже великий розмір вводу, пов'язаний із зображенням (кожен піксель є відповідною змінною), навіть у структурі поверхні (на відміну від глибоких структур) знадобиться дуже велика кількість нейронів. Наприклад, вага повністю клейкого шару (невеликого) зображення розміром 100×100 становить 10000. Операція згортки вирішує цю проблему, оскільки зменшує кількість доступних

параметрів, що робить мережу з меншою кількістю параметрів глибшою. Наприклад, незалежно від розміру зображення, на кожен 5×5 бруковку з однаковою загальною вагою потрібно лише 25 вільних параметрів. Тому він вирішує проблему зникнення або вибуху градієнта в традиційному багатошаровому тренуванні нейронної мережі з багатьма шарами за допомогою зворотного поширення..

ЗНМ можуть включати локальні чи глобальні шари агрегації, може поєднувати вихід шару кластерів нейронів до виходу наступного шару нейронів. Наприклад, максимізаційне агрегування використовує максимальне значення кожного кластера нейронів у попередньому шарі. Іншим прикладом є середній пул, який використовує середнє значення кожного кластера нейронів у попередньому шарі.

Повністю з'єднаний шар з'єднує кожен нейрон в одному шарі з кожним нейроном у наступному шарі. В принципі це те саме, що і традиційна нейронна мережа багатошарового перцептрона.

ЗНМ використовує загальні ваги в згортковому шарі, це означає, що для кожного поля прийому шару використовується один і той же фільтр (банк ваг); це зменшує необхідну кількість пам'яті та покращує продуктивність. [8]

1.6. Огляд аналогів для розроблюваної системи та постановка задачі.

На сьогодні використання штучного інтелекту у медицині стрімко поширюється. Кожна країна світу пропонує власні розробки та методи для застосування зокрема нейронних мереж. Але кожна існуюча система має обмеження та особливості, з якими може зіткнутися медична клініка, яка забажає скористатися нейронними мережами.

Програмне забезпечення ScanNav Audit забезпечує підтримку в реальному часі для акушерських ультразвукових лікарів, які виконують сканування аномалії на 20 тижні вагітності. ScanNav Audit спрямований на те, щоб забезпечити відповідність повного набору скануючих зображень за призначенням та відповідності необхідному протоколу сканування при зйомці під час процедури. Програмне забезпечення також забезпечує запис результатів роботи кожного сонографа, дозволяючи керівникам контролювати персонал та формувати частину вимог ведення обліку в клініці.

Програмне забезпечення AutoCapture ScanNav автоматично фіксує та аналізує всі площини ультразвукового зображення в режимі реального часу, коли сонограф переміщує ультразвуковий зонд над абоменом пацієнта під час сканування аномалії плоду протягом 20 тижнів. Поточна версія програмного забезпечення потім автоматично вибирає та зберігає ключові зображення, необхідні для відповідності протоколу FASP у Великобританії. [9]

Щоб навчати алгоритми нейронних мереж, розробник співпрацює з провідними медичними установами, щоб створити великі бібліотеки реальних ультразвукових зображень, які охоплюють якомога більшу кількість анатомічних варіантів. Це дозволяє моделям AI, які створює розробник, витягувати помітні риси, що дозволяють їм відрізнити нормальне від аномального.

Оскільки ультразвук є властивою галасливою модальністю зображень, ми розробили ряд методів обробки зображень для підвищення продуктивності.

Поєднання цих факторів - розмір бібліотеки зображень та досвід клінічного колективу розробника забезпечують бездоганні досягнення у створенні продуктів, що підтримують прийняття клінічних рішень. Вони не тільки скоротять час сканування, але й розширять використання ультразвуку, надаючи неспеціалістам необхідну підтримку.

Головними особливостями точності такої розробки є навчання моделі по великій базі зображень та об'єднання досвіду різних медичних клінік для

					ІАЛЦ.045490.004 ПЗ	Арк.
						15
Змін.	Арк.	№ докум.	Підпис	Дата		

однієї мети. Основним недоліком навчання нейронних мереж є високі вимоги до апаратної частини комп'ютеру. Клініка, яка бажає користуватися подібними продуктами повинна гарантувати спеціалізоване обладнання – комп'ютери з особливими відеоадаптерами та процесорами. У випадку відсутності можливості гарантувати таке обладнання клініка не зможе користуватися таким прогресивним засобом. Але на сьогодні використовувати власне обладнання не є обов'язковим. Сучасні WEB хмарні сервіси надають можливість навчати нейронні мережі в умовах обмеженості ресурсів. Британські розробки не можуть бути застосовані на хмарних сховищах.

Другим основним недоліком Британської розробки є розповсюдження продукту лише на території Об'єднаного Королівства. Таким чином українські клініки не мають доступу до таких розробок.

Для розв'язання виявлених недоліків перед розробкою модуля розпізнання зображень УЗД було поставлено наступні задачі:

- 1) Розробити програмний продукт, який зможе надати можливість науковцям навчати згорткову нейронну мережу по уже напрацьованій базі;
- 2) Надати функціонал створення бази даних вхідних зображень із архіву зображень УЗД;
- 3) Розробити можливість розпізнання об'єктів на медичних зображеннях засобами уже навченої моделі;
- 4) Надати можливість аналізувати результати виконання розпізнавання засобами модуля;
- 5) Вбудувати функціонал використання хмарних сховищ в якості апаратних потужностей для навчання моделі;
- 6) Забезпечити високу стабільність виконання програмного забезпечення, розробити довідку для користувача, інформувати користувача про важливу інформацію спеціальними сповіщеннями.

ВИСНОВКИ ДО РОЗДІЛУ 1

У даному розділі описано предметну область розпізнавання зображень. Проведено аналіз аномалій внутрішньоутробного розвитку, означено терміни скринінгу плоду та описано дефекти, що можуть бути знайдені під час УЗД скринінгу. Проведено аналіз медичних зображень, їх видів та способів отримання таких зображень.

У розділі розглядається штучний інтелект як сучасний засіб розв’язання задачі автоматизованого розпізнавання зображень. Проаналізовано засоби машинного навчання штучного інтелекту, проаналізовано алгоритми навчання нейронних мереж як різновид штучного інтелекту. Детально було розглянено згорткові нейронні мережі як пристосований засіб для розпізнання зображень.

У розділі описано аналог для розроблюваної системи – британську розробку ScanNav Audit. Описано особливості та переваги цієї системи. Також було виявлено недоліки та обмеження у розробки. На основі недоліків з метою їх вирішення було сформовано задачі, які мають бути розв’язані у розроблюваному модулі.

					ІАЛЦ.045490.004 ПЗ	Арк.
						17
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

2.1. Вибір мови програмування

Машинне навчання можливо реалізувати за допомогою більшості відомих мов програмування. До таких мов програмування відносять Java, JavaScript, C++, R, MathLab, Python, тощо. Кожна із мов програмування має власні переваги та недоліки. Серед цих мов було обрано Python.

Основними перевагами є:

- Чистий синтаксис (для вибору блоків слід використовувати відступи);
- портативність програми (для більшості інтерпретованих мов це унікально);
- стандартна дистрибуція має велику кількість корисних модулів (включаючи модулі для розробки графічних інтерфейсів);
- Можливість використання Python в режимі діалогу (корисно для експериментів та простих питань);
- Стандартний розподіл має просте, але потужне середовище розробки, що називається IDLE, написане на Python;
- Зручний для вирішення математичних задач (використовуючи метод обробки складних чисел, може обробляти цілі числа будь-якого розміру, може використовуватися як потужний калькулятор у діалоговому режимі)
- Відкритий код (функції, відредаговані іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний метод об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типу та інтерпретована мова роблять його ідеальним для написання сценаріїв та швидкий розвиток додатків у багатьох галузях промисловості на більшості платформ.

Інтерпретатор Python та багата стандартна бібліотека (вихідний код та бінарні дистрибутиви всіх основних операційних систем) можна отримати на

веб-сайті Python www.python.org і можуть розповсюджуватися безкоштовно. Цей же сайт має випуски та посилання на багато модулів, програм, утиліт та інших документів.

Інтерпретатор Python може бути розширений за допомогою функцій та типів даних, розроблених на C або C++ (або іншій мові, яку можна викликати з C). Python також може використовуватися як мова розширення для додатків, які потребують подальшої налагодження.

Python підтримує динамічне введення тексту, тобто тип змінних визначається лише під час виконання. У основних типах слід подбати про підтримку цілих чисел довільної довжини та складних чисел. У Python є багата бібліотека для обробки рядків, особливо рядків, що використовують кодування Unicode.

З колекцій Python підтримує кортежі, списки (масиви), словники (асоціативні масиви) та колекції у версії 2.4.

Класова система підтримує багаторазове успадкування та метапрограмування. Будь-який тип (включаючи базові типи) є частиною класової системи, а при необхідності навіть може успадкувати базові типи.

Багата бібліотека стандартів - одна з привабливих особливостей Python. Існують інструменти для обробки багатьох мережевих протоколів та Інтернет-форматів, наприклад, для запису HTTP-серверів та клієнтів, для розбору та створення електронних листів, модулів для XML тощо. Набір модулів для операційної системи дозволяє писати кросплатформенні програми. Існують модулі для обробки регулярних виразів, кодування тексту, мультимедійних форматів, протоколів шифрування, файлів, серіалізації даних, тестування одиниць тощо.

Python та більшість бібліотек безкоштовні і включаються у вихідний код. Більше того, на відміну від багатьох відкритих систем, ця ліцензія не обмежує використання Python в комерційній розробці і не покладає ніяких інших зобов'язань, крім авторських прав. [10]

2.2. Архітектура R-CNN.

Для реалізації навчання нейронної мережі було обрано архітектуру R-CNN, алгоритм Faster R-CNN.

Згорткові нейронні мережі (CNN) в машинному навчанні - це тип прямого поширення глибокої штучної нейронної мережі, який успішно застосовується для візуального аналізу зображень.

Завдяки використанню спеціальної операції - фактичної згортки - ви можете одночасно зменшити кількість інформації, що зберігається в пам'яті, щоб краще впоратися із зображеннями з більшою роздільною здатністю та виділити підтримуючі функції зображення, такі як краї, Силует або обличчя. На наступному рівні обробки цих країв та граней можуть бути ідентифіковані повторні фрагменти текстури, і ці фрагменти можуть бути надалі розроблені у фрагменти зображення.

По суті, кожен шар нейронної мережі використовує своє власне перетворення. Якщо мережа функціонує на першому шарі з такими поняттями, як "край" та "обличчя", далі будуть використані поняття "текстура" та "частина об'єкта". Після такої обробки ми можемо правильно класифікувати зображення або вибрати потрібний об'єкт на зображенні на завершальній стадії. [12]

Щоб обійти проблему вибору величезної кількості областей, Рос Гіршик та ін. запропонували метод, при якому ми використовуємо вибіркового пошуку, щоб витягти з зображення лише 2000 областей, і він назвав їх регіональними зонами інтересу. Тому зараз, замість того, щоб класифікувати величезну кількість регіонів, можна працювати з 2000 регіонами. Ці 2000 регіональних пропозицій генеруються з використанням алгоритму вибіркового пошуку:

1. Створити початкову підсегментацію, згенерувати багато регіонів-кандидатів;

2. Використовувати жадібний алгоритм для рекурсивного об'єднання подібних регіонів у великі;

3. Використовувати згенеровані регіони для створення кінцевих пропозицій регіону-кандидата.

Ці 2000 пропозицій регіонів-кандидатів викривляються у квадрат і надходять у згортку нейронної мережі, яка виробляє 4096-мірний вектор ознак як вихідний. ЗНМ виступає в якості екстрактора ознак і вихідний щільний шар складається з функцій, витягнутих з зображення, і витягнуті ознаки подаються в SVM для класифікації присутності об'єкта в цій пропозиції регіону-кандидата. На додаток до передбачення присутності об'єкта в межах пропозиції регіону, алгоритм також передбачає чотири значення, які є значеннями зміщення для підвищення точності обмежувального регіону. Наприклад, з урахуванням пропозиції регіону, алгоритм передбачав би присутність людини, але обличчя цієї особи в межах цієї пропозиції регіону можна було б скоротити наполовину.

Основні проблеми алгоритму R-CNN:

- Треба багато часу для навчання мережі, оскільки доводиться класифікувати 2000 регіонів по кожному зображенню.

- Метод не може бути реалізовано в реальному часі, оскільки для кожного тестового зображення потрібно близько 47 секунд.

- Алгоритм вибіркового пошуку є фіксованим алгоритмом. Тому на даному етапі навчання не відбувається. Це може призвести до формування поганих пропозицій регіонів-кандидатів.

Автор попередньої роботи (R-CNN) вирішив деякі недоліки R-CNN для побудови більш швидкого алгоритму виявлення об'єктів, що називається швидким R-CNN (Fast R-CNN). Підхід аналогічний алгоритму R-CNN. Але, замість того, щоб подавати регіональні пропозиції до ЗНМ, ми подаємо до мережі вхідне, щоб створити згорткову карту властивостей. На карті згорткових характеристик ми ідентифікуємо області пропозицій і деформуємо

їх у квадрати і використовуючи шар об'єднання (RoI), ми їх змінюємо у фіксований розмір, щоб він міг надходити у повністю пов'язаний шар (Fully connected layer). З вектора особливості RoI ми використовуємо шар softmax для прогнозування класу запропонованої області, а також значень зміщення для обмежувального регіону.

Причина того, що «швидкий R-CNN» швидше, ніж R-CNN, в тому, що ви не треба щоразу подавати пропозиції регіону до згорткових нейронних мереж. Замість цього операція згортки виконується тільки один раз на зображення, і з неї генерується карта об'єктів.

Обидва вищезгадані алгоритми (R-CNN & Fast R-CNN) використовують вибіркового пошуку, щоб знайти пропозиції регіону. Вибірковий пошук є повільним і трудомістким процесом, що впливає на продуктивність мережі. Тому Shaoqing Ren et al. придумав алгоритм виявлення об'єктів, який виключає алгоритм вибіркового пошуку і дозволяє мережі вивчати пропозиції регіону.

Подібно до швидкого R-CNN, зображення надається в якості вхідного сигналу для згорткової мережі, яка забезпечує згортку. Замість використання алгоритму вибіркового пошуку на карті об'єктів для ідентифікації пропозицій регіону використовується окрема мережа для прогнозування пропозицій регіону. Прогнозовані області пропозицій потім змінюють, використовуючи шар об'єднання RoI, який потім використовують для класифікації зображення в межах запропонованої області і прогнозування значень зміщення для обмежувальних полів. [11]

2.3. Засоби мови програмування Python для машинного навчання.

Detecto - це пакет Python, який дозволяє будувати повністю функціонуючі моделі комп'ютерного зору та виявлення об'єктів. Функціями Detecto є: висновки щодо нерухомих зображень та відео, передача навчання на користувальницьких наборах даних та серіалізація моделей у файли. Detecto використовує Faster R-CNN ResNet-50 FPN алгоритм із пакету PyTorch. [13]

PyTorch - бібліотека машинного навчання для мови Python з відкритим вихідним кодом, створена на базі Torch. Використовується для вирішення різних завдань: комп'ютерний зір, обробка природної мови. Створена ця бібліотека на базі бібліотеки Torch для Lua. Ці пакети надають велику кількість алгоритмів для машинного навчання. [14]

2.4. Засіб для створення інтерфейсу.

Для створення графічного інтерфейсу модуля було використано бібліотеку PyQt5. PyQt5 є одним з найбільш часто використовуваних модулів для створення GUI додатків в Python. PyQt5 реалізований у вигляді набору python-модулів. Ця бібліотека має понад 620 класів і 6000 функцій і методів.

Основні можливості PyQt5:

- Існуючий набір віджетів графічного інтерфейсу;
- стилі віджетів;
- доступ до баз даних за допомогою SQL (ODBC, MySQL, PostgreSQL, Oracle);
- QScintilla, заснований на Scintilla віджет текстового редактора;
- підтримку інтернаціоналізації (i18n);

- парсер XML;
- підтримку SVG;
- інтеграцію з WebKit, двигуном рендеринга HTML;
- підтримку відтворення відео і аудіо.

PyQt також включає в себе Qt Designer (Qt Creator) - дизайнер графічного інтерфейсу користувача. Програма руіс генерує Python код з файлів, створених в Qt Designer. Це робить PyQt дуже корисним інструментом для швидкого прототипування. Крім того, можна додавати нові графічні елементи управління, написані на Python, в Qt Designer.

Qt Designer дозволяє створювати графічні інтерфейси користувача за допомогою ряду інструментів. Існує панель інструментів «Панель віджетів», в якій доступні для використання елементи інтерфейсу - віджети, такі як, наприклад, «випадаючий список» ComboBox, «поле введення» LineEdit, «кнопка» PushButton і багато інших. Кожен віджет має свій набір атрибутів, який визначається відповідним класом бібліотеки Qt. Властивості віджета можна змінити за допомогою "Редактора властивостей". Кожен клас атрибутів віджетів має спеціальний редактор. [15]

2.5. Засіб для створення аннотацій LabelImage.

Машинне навчання потребує бази даних зображень із аннотаціями в якості вхідних даних. Вхідні зображення для розроблюваного модуля це медичні зображення УЗД. Модель нейронної мережі під час навчання аналізує не тільки зображення, але і об'єкти, що на них присутні. Ці об'єкти оформлюються у вигляді аннотацій. Аннотації це XML файл, що містить в собі назву об'єкту, розміщення, та іншу спеціалізовану інформацію. Аннотації зберігаються у форматі Pascal VOC.

					ІАЛЦ.045490.004 ПЗ	Арк.
						24
Змін.	Арк.	№ докум.	Підпис	Дата		

Приклад фрагменту анотації зображено на рисунку 2.1.

```
<annotation>
  <folder>images</folder>
  <filename>image0.jpg</filename>
  <path>E:\images\image0.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>400</width>
    <height>310</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>child</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>29</xmin>
      <ymin>134</ymin>
      <xmax>326</xmax>
      <ymax>282</ymax>
    </bndbox>
  </object>
  .
</annotation>
```

Рисунок 2.1 – Фрагмент анотації до зображення.

2.6. Засіб для хмарної обробки даних Google Colab.

Google Colab - це форк популярного середовища Jupyter notebook. Це середовище дає безкоштовний доступ до ноутбуків через google drive в .ipynb форматі і запускати програми можна локально. Підтримується Python 2.7 і 3.6. Код виконується на сервері в docker контейнері. Можна закрити браузер, всі процеси на сервері продовжать працювати, пізніше можна підключитися до сервера знову. Docker контейнер видається користувачеві у тимчасове користування на 12 годин. Користувач має root привілеї, і може встановлювати і запускати будь-які програми всередині контейнера. Colaboratory (далі colab) також підтримує спільну роботу над ноутбуком, за типом google docs.

					ІАЛЦ.045490.004 ПЗ	Арк.
						25
Змін.	Арк.	№ докум.	Підпис	Дата		

Project Jupyter - це некомерційна організація, створена для "розробки програмного забезпечення з відкритим кодом, відкритих стандартів та служб для інтерактивних обчислень на десятках мов програмування". Відключення від IPython у 2014 році Фернандо Переза, Project Jupyter підтримує середовища виконання декількома десятками мов. Ім'я проекту Юпітера - це посилання на три основні мови програмування, які підтримує Юпітер, а саме: Julia, Python і R, а також вшановує пам'ятки Галілея, що фіксують відкриття лун Юпітера. Проект Jupyter розробив та підтримує інтерактивні обчислювальні продукти Jupyter Notebook, JupyterHub та JupyterLab, версію Jupyter Notebook нового покоління. [16]

Філософія проекту Jupyter полягає у підтримці інтерактивної науки про дані та наукових обчислень на всіх мовах програмування шляхом розробки програмного забезпечення з відкритим кодом. За даними веб-сайту Project Jupyter, "Jupyter завжди буде 100% програмним забезпеченням з відкритим кодом, безкоштовним для використання та випущеним за ліберальними умовами модифікованої ліцензії BSD".

					ІАЛЦ.045490.004 ПЗ	Арк.
						26
Змін.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

У розділі сформовано перелік засобів, за допомогою яких можливо реалізувати програмний продукт, який задовольняє умови завдання, поставлені у пункті 1.

Обґрунтовано особливості та переваги мови програмування Python, алгоритму навчання нейронної мережі. Здійснено пошук бібліотек для мови програмування Python, які виконуватимуть функцію навчання нейронної мережі – Detecto на базі PyTorch. Проаналізовано знайдені бібліотеки та виявлено їх переваги. Обрано засіб для розробки інтерфейсу програмного забезпечення – QtDesigner 5, обґрунтовано переваги обраного засобу. В якості інструмента для створення вхідної бази даних зображень обрано засіб LabelImg – в розділі розглянено можливості засобу.

У розділі описано засіб хмарного сховища як додаткове джерело апаратних потужностей для навчання нейронних мереж.

					ІАЛЦ.045490.004 ПЗ	Арк.
						27
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Розробка архітектури модуля розпізнавання зображення УЗД.

Архітектура програмного забезпечення - це структура програми або обчислювальної системи, яка включає програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними. Документування архітектури ПЗ спрощує процес комунікації між зацікавленими особами, дозволяє зафіксувати прийняті на ранніх етапах проектування рішення про високорівневий дизайн системи і дозволяє використовувати компоненти цього дизайну і шаблони повторно в інших проектах.

Архітектура ПЗ включає в себе:

- 1) вибір структурних елементів і їх інтерфейсів, за допомогою яких складена система, а також їх поведінки в рамках співпраці структурних елементів;
- 2) з'єднання обраних елементів структури і поведінки у все більш крупні системи;
- 3) архітектурний стиль, який направляє всю організацію - всі елементи, їх інтерфейси, їх співпраця і їх з'єднання.

Розроблений модуль для розпізнавання зображення УЗД призначений для використання людиною-користувачем – науковим працівником медичного закладу. Користувач є першим структурним елементом модуля. Користувач приймає рішення щодо вибору мети використання модулю – навчання нової моделі, чи здійснення дослідження за допомогою уже готової моделі; вибору вхідних даних для аналізу; приймає рішення щодо апаратних засобів, що здійснюють дослідження – власні засоби, чи хмарні; приймає рішення щодо кінцевого результату аналізу.

Розроблений модуль передбачає широкий функціонал та обширне

					ІАЛЦ.045490.004 ПЗ	Арк.
						28
Змін.	Арк.	№ докум.	Підпис	Дата		

поле для прийняття рішень кінцевим користувачем. Для ефективного використання модулю необхідним буде наступний структурний елемент – інтерфейс користувача. Саме за допомогою користувацького інтерфейсу є можливість прийняття рішень, вибору вхідних даних та отримання вихідних даних. Користувацький інтерфейс є другим структурним елементом архітектури.

Третім структурним елементом є обчислювально-статичний блок. До цього елемента входять усі обчислювальні засоби, які мають відношення до обробки даних, прийняття рішень та виводу результатів. Цей структурний елемент включає в себе розроблений програмний продукт та нейронна мережа, що здійснює аналіз.

Четвертим та останнім структурним елементом є блок збереження даних. Розроблений модуль не передбачено встановлювати у локальній мережі чи глобальній мережі Інтернет. Модуль використовує у якості вхідних даних файли, що зберігаються на локальному диску пристрою користувача за допомогою файлової системи Windows, збереження даних також відбувається на локальному диску пристрою користувача. Структурним елементом передбачається файлова система пристрою.

Згідно із здійсненим вибором можна побудувати схему архітектури програмного забезпечення. Побудована схема представлена в додатку А.

3.2. Розробка користувацького інтерфейсу.

Інтерфейс користувача — інтерфейс, що забезпечує передачу інформації між користувачем-людиною і програмно-апаратними компонентами комп'ютерної системи.

Інтерфейс користувача, в галузі промислового дизайну, взаємодії між людиною та комп'ютером — це простір, де відбувається співпраця між

людьми та машинами. Мета цієї взаємодії, полягає у тому, щоб забезпечити досконалу роботу та керування машиною з боку людини, а машина одночасно, надає інформацію, яка допомагає прийняттю рішень операторами. Прикладами цієї широкої концепції інтерфейсів користувача, є: інтерактивні аспекти комп'ютерних операційних систем, ручні інструменти, операторські засоби керування важким обладнанням та елементи контролю над технологічними процесами. Конструктивні міркування, що застосовуються під час створення інтерфейсів користувача, пов'язано з такими галузями, як ергономіка та психологія.

Мета створення інтерфейсу користувача, полягає у тому, щоб розробити інтерфейс користувача, який спрощує процес керування програмою.

У розробленому модулі під час першого запуску програма має пропонувати користувачу здійснити вибір – користувач працюватиме із уже навченою нейронною мережею, чи бажає навчити нову. Також користувачу пропонується ознайомитися із довідковою інформацією. На рисунку 3.1 представлено скріншот форми, що буде запущено.

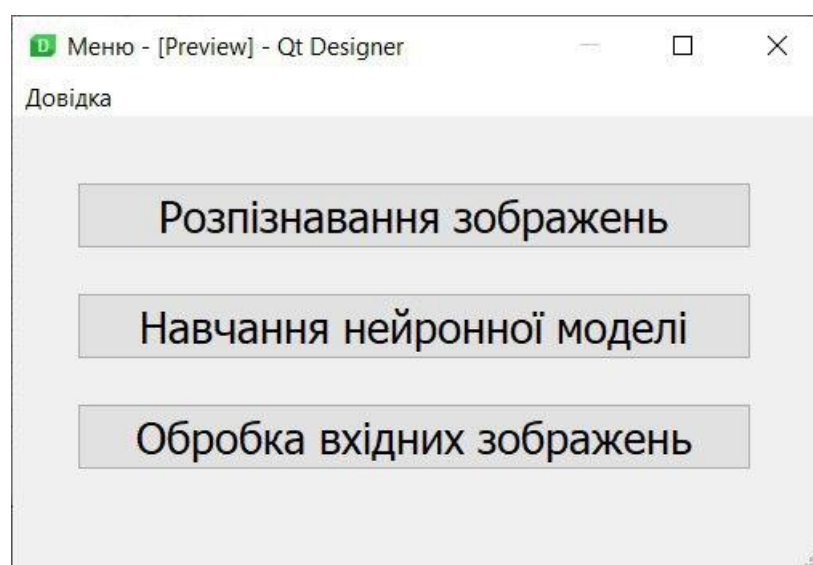


Рисунок 3.1 – Перший запуск програми.

Елементи типу «Кнопка» запускають окремі вікна для задач «Розпізнавання зображень» або «Навчання моделі». У рядку меню

представлена можливість запуску довідки. Рядок меню присутній на усіх формах програми. Вихід із програми передбачено за допомогою стандартної кнопки ОС Windows «Закрити».

На рисунку 3.2 представлено форму, що буде запущено натисканням кнопки «Розпізнавання зображень». Представлений інтерфейс дозволяє досліджувати зображення за допомогою уже навченої нейронної мережі.

Рисунок 3.2 – запуск форми «Розпізнавання зображень».

Представлену форму можна поділити на декілька умовних частин.

Перша частина представлена заголовком «Модель, яка буде використана для розпізнання», рядком для відображення шляху до файлу, та кнопкою «Обрати модель». Кнопка «Обрати модель» викликає вибір файлу нейронної

мережі у файловій системі ОС Windows. Після вибору файлу у рядку зліва буде відображено шлях до файлу.

Друга частина представлена заголовком «Зображення, що буде розпізнано», кнопкою «Обрати зображення» та рядком для відображення

шляху до файлу. Кнопка «Обрати зображення» викликає вибір файлу зображення, яке користувач бажає дослідити. Файл обирається засобами файлової системи ОС Windows. Після вибору файлу у рядку зліва відображається шлях до файлу.

Третя частина форми представлена заголовком «Результат буде збережено», кнопку «Обрати теку» та рядок для відображення шляху до теки. Кнопка «Обрати теку» дозволяє у файловій системі ОС Windows задати шлях для збереження результатів. Після вибору теки шлях буде відображено у рядку зліва.

Четверта частина форми представлена заголовком «Список об'єктів для виявлення», кнопкою «Імпортувати список об'єктів», полем для ручного введення об'єктів, полем для відображення списку заданих об'єктів, кнопками «Додати» та «Вилучити». У четвертій частині форми користувач обирає які об'єкти має бути розпізнано на зображенні. Назву об'єкту можна ввести вручну у спеціальному полі та натиснути «Додати». Після натискання кнопки назва об'єкту з'являється у спеціальному полі. Кнопка «Імпортувати список об'єктів» дозволяє зчитати список усіх необхідних об'єктів із файлу. Для вилучення об'єкту із аналізу необхідно натиснути на нього у списку та натиснути кнопку «Вилучити».

На рисунку 3.3 представлено форму, на якій користувач здійснив усі дії, необхідні для запуску розпізнавання.

Перед запуском програми користувач може встановити чекбокс «Показати результат виконання» запускає зображення, на якому нейронна мережа розпізнала об'єкти. Зображення запускається засобами розробленого

модуля. Кнопка «Виконати пошук об’єктів» запускає аналіз зображення. На рисунку 3.4 представлено результат розпізнавання.

Програма для розпізнавання зображень УЗД

Довідка Модель

Модель, яка буде використана для розпізнання:

C:\Users\Oleksandr\source\repos\detecto\model_weights.pth

Обрати модель

Зображення, що буде розпізнано:

E:\111.jpg

Обрати зображення

Результат буде збережено:

E:\result.png

Обрати теку

Список об’єктів для виявлення:

child
body
head

+

Введіть назву об’єкту

Імпортувати список об’єктів

-

☒ Показати результат після виконання

Виконати пошук об’єктів

Рисунок 3.3 – Заповнена форма «Розпізнавання зображень»

На рисунку 3.4 представлено виведення результату розпізнавання зображення.

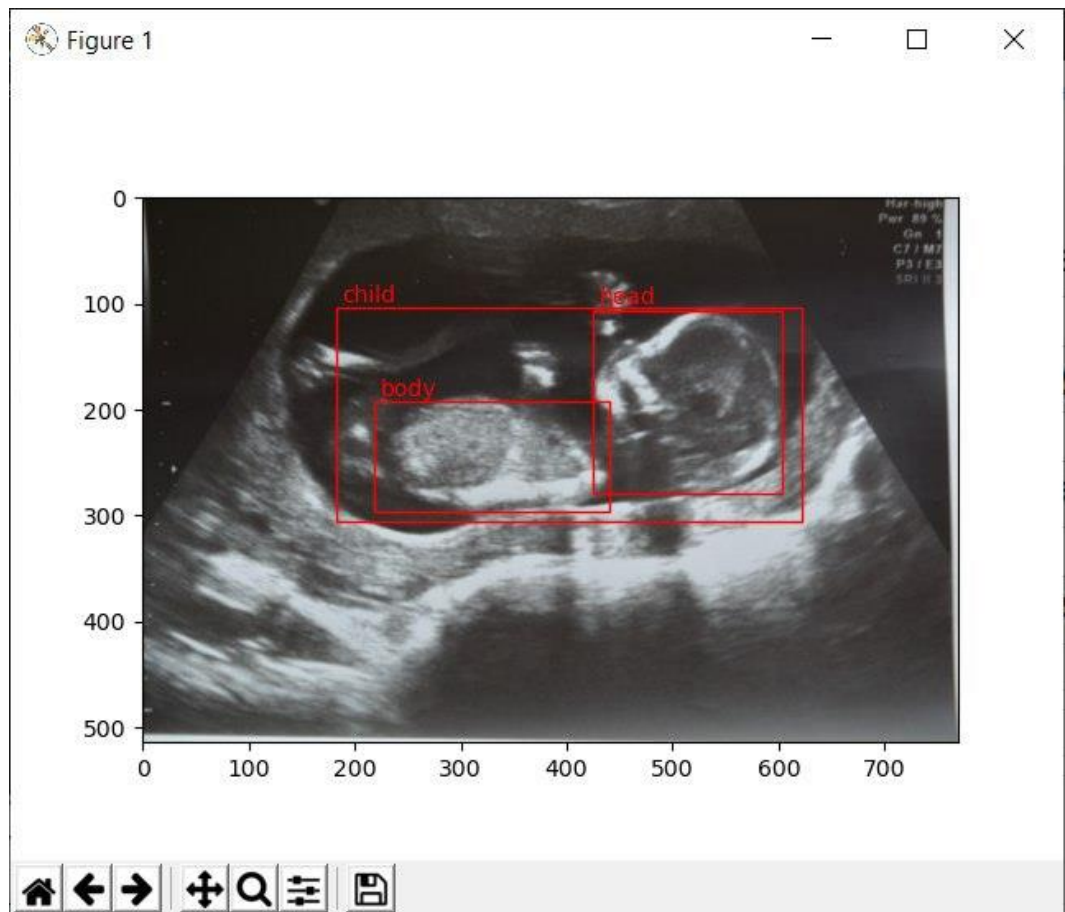


Рисунок 3.4 – Результат виконання розпізнавання зображення.

На рисунку 3.5 представлено форму, що буде запущено натисканням кнопки «Навчання моделі».

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

34

Рисунок 3.5 – Форма «Навчання нової моделі»

Форма «Навчання нової моделі» використовується для створення нової моделі нейронної мережі. Функціонально форма поділяється на декілька частин.

Перша частина містить заголовок «Тека, в якій зберігаються дані, для навчання моделі», кнопку «Обрати теку» та поле для відображення шляху до теки. За допомогою кнопки «Обрати теку» користувач встановлює з якої теки буде зчитано вхідні дані до програми.

Друга частина містить заголовок «Результат буде збережено», кнопку обрати теку та поле для відображення шляху до теки. За допомогою кнопки

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

35

«Обрати теку» користувач встановлює в якій теці зберігатиметься нова модель.

Третя частина містить заголовок «Список об'єктів для виявлення», поле для введення назви об'єкту, кнопку «Додати», кнопку «Імпортувати список об'єктів», поле для відображення списку об'єктів, що визначатиме модель, кнопку «Вилучити», заголовок «Використання потужностей» та перемикачі «Власного комп'ютера» та «Google Colab». Додавати назви об'єктів можливо двома способами – вказати назву об'єкту у відповідному полі та натиснути кнопку «Додати». Далі об'єкт з'являється у списку. Вилучити об'єкт із списку можливо обравши об'єкт та натиснувши кнопку «Вилучити». Кнопка «Імпортувати список об'єктів» дозволяє завантажити список назв об'єктів із текстового файлу.

Модуль розпізнавання об'єктів надає можливість користувачеві обирати де саме будуть виконуватися обчислювання – на пристрої користувача, чи засобами Google Colab. Для здійснення вибору користувач має обрати перемикач «Власного комп'ютера» чи «Google Colab».

На рисунку 3.7 представлено представлено форму, на якій користувач здійснив усі дії, необхідні для запуску навчання. На рисунку 3.6 представлено сповіщення про завершення навчання моделі.

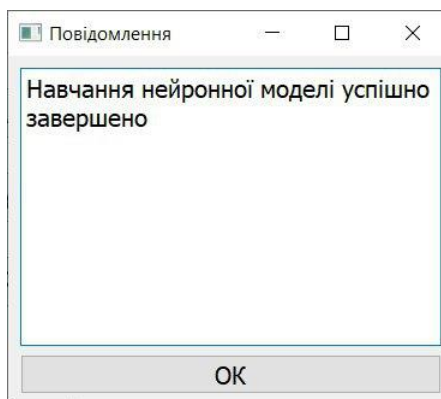


Рисунок 3.6 – Повідомлення про завершення навчання моделі.

Навчання нової моделі

Довідка

Тека, в якій зберігаються дані, для навчання моделі

E:\images

Обрати теку

Результат буде збережено:

E:\

Обрати теку

Список об'єктів для виявлення:

child
body
head

+

Введіть назву об'єкту

Імпортувати список об'єктів

-

Використання потужностей

☒ Власного комп'ютера

☐ Google Coolab

Створити модель

Скористайтесь довідкою, щоб підготувати дані для навчання моделі

Рисунок 3.7 – Заповнена форма «Навчання нової моделі».

Для навчання нової моделі необхідно обробити вхідні зображення. Навчання моделі передбачає читання зображень та встановлених на них об'єкти. Встановлювати об'єкти для розпізнавання має кваліфікована людина. Встановлення коректних об'єктів знижує можливість похибки під час розпізнавання. Виклик обробки вхідних зображень відбувається із головного меню натисканням кнопки «Обробка вхідних зображень». На рисунку 3.8. представлено інтерфейс обробки зображень.

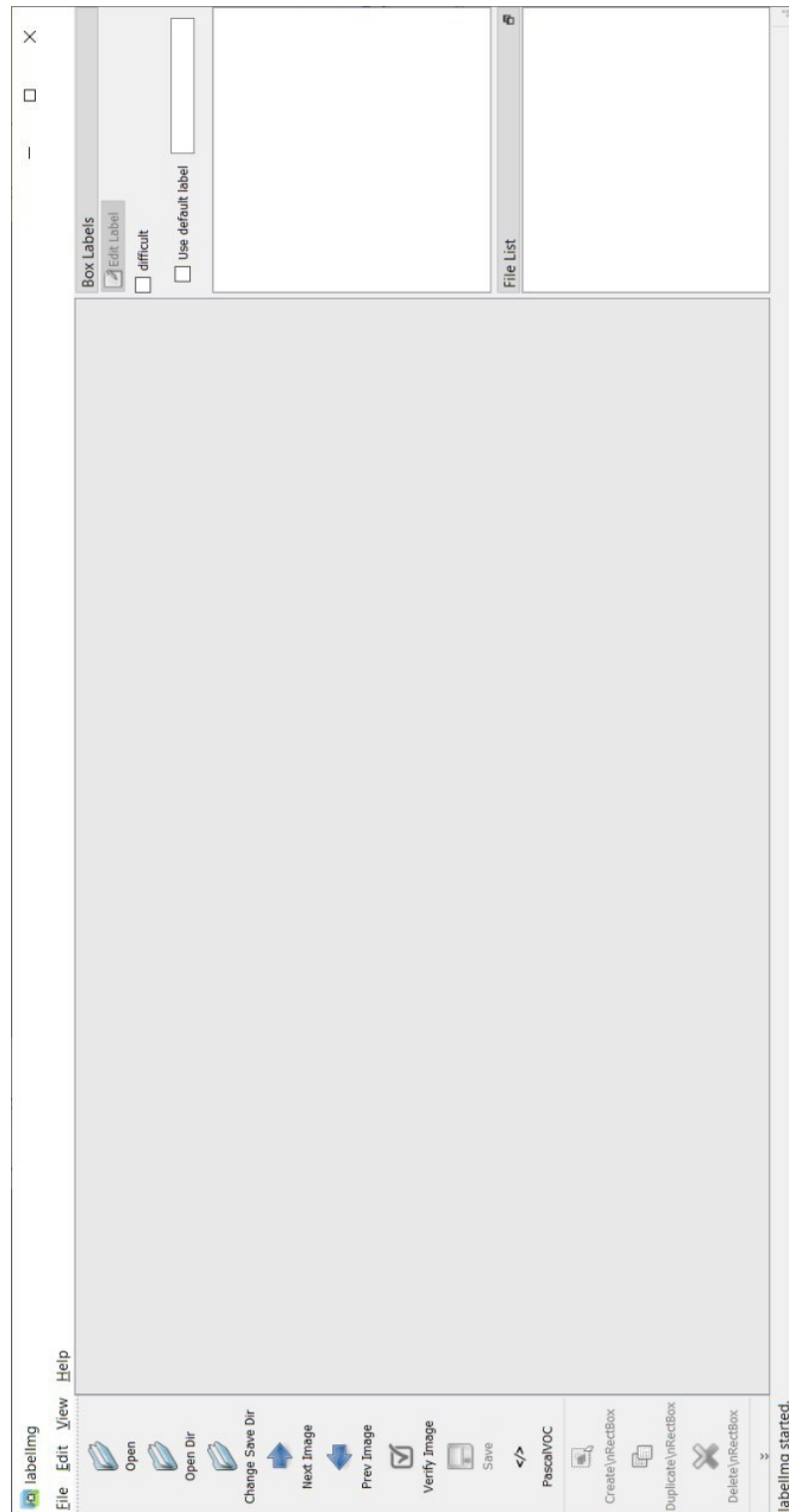


Рисунок 3.8 – Інтерфейс обробки зображень.

Даний інтерфейс є безкоштовним модулем мови програмування Python LabelImage. Вибір вхідних даних можливий кнопками Open – обирає файл у

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

38

файловій системі, або OpenDir – обирає усі зображення в теці за допомогою файлової системи. На рисунку 3.9 представлено процес задання об’єктів.



Рисунок 3.9 – Процес встановлення об’єктів.

Змін.	Арк.	№ докум.	Підпис	Дата

Для встановлення об'єктів необхідно натиснути кнопку «Create RectBox» та виділити необхідний об'єкт на зображенні. Після виділення об'єкту модуль LabelImage запитає назву об'єкту – вікно представлено на рисунку 3.10.



Рисунок 3.10 – Встановлення назви об'єкту.

Важливо встановлювати однакову назву для однакових об'єктів – це знижує можливість помилки під час розпізнавання зображення моделлю нейронної мережі. Після натискання кнопки «OK» назва об'єкту з'являється у списку об'єктів.

Після встановлення назви необхідно натиснути кнопку «Save». Кількість ітерацій дорівнює кількості зображень. Оброблені зображення слід зберігати у одній теці.

3.3. Засоби реалізації програмного коду.

Для реалізації програмного продукту застосовано об'єктно орієнтовану парадигму. Це така парадигма, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. В програмі спочатку створюється клас, який містить поля для збереження інформації, та методи для реалізації програми.

У представленому модулі створено 8 класів – кожен вид функціоналу представлений у виді окремого класу. Діаграма класів представлено в Додатку Б.

Графічний інтерфейс модуля реалізовано класами `UI_ModelWindow`, `UI_MessageWindow`, `UI_ProgramWindow`, `UI_MenuWindow`. Полями цих класів є елементи інтерфейсу, такі як: кнопка, поле для введення даних, заголовки, тощо. У кожного класу є метод, який встановлює початкові значення для полів. Класи є абстрактними – об'єкти цих класів не створюються.

Методами класу `MenuWindow` реалізовано початкове меню. Клас `MenuWindow` успадковує клас `UI_MenuWindow` – таким чином реалізовано об'єднання графічної та програмної частини функціоналу головного меню. У даного класу є три методи, кожен із них створює об'єкти інших класів та викликає їх методи. Таким чином реалізовано запуск функціоналу навчання нової моделі, аналізу зображення та створення анотацій.

Методами класу `ModelWindow` реалізовано функціонал навчання нової моделі. Клас `ModelWindow` успадковує клас `UI_ModelWindow` – таким чином реалізовано об'єднання графічної та програмної частини функціоналу навчання нової моделі. Методом `set_data()` реалізовано вибір теки з вхідними зображеннями. Шлях до обраної теки зберігається у полі `datapath`. Методом `set_result_model()` реалізовано вибір теки для збереження результату – навченої моделі нейронної мережі. Шлях зберігається у полі `resultmodelpath`. За допомогою методу `import_classes()` реалізовано імпорт назв об'єктів із текстового файлу. Зберігаються дані у полі `classes`. Методами `add_class()` реалізовано ручне додання назви об'єкту, що введено до спеціального поля. Метод `delete_class()` дозволяє вилучити назву об'єкту із списку, в якому обирається необхідний об'єкт. Метод `make_model()` запускає процес навчання моделі, враховуючи вибір потужностей власного комп'ютера, або Google Colab. У випадку, якщо обрано засіб Google Colab модулем створюється

локальний сервер та надсилається запит на виконання програмного коду на Google Colab.

Методами класу ProgramWindow реалізовано функціонал розпізнавання об'єктів на зображенні. Клас ProgramWindow успадковує клас UI_ProgramWindow – таким чином реалізовано об'єднання графічної та програмної частини функціоналу розпізнавання об'єктів на зображенні. Шлях до моделі нейронної мережі, яку буде використано встановлюється методом set_model(), зберігається шлях у полі modelpath. Шлях до зображення, на якому розпізнається об'єкт встановлюється методом set_image(), зберігається шлях у полі imagepath. Шлях до теки, в якій буде збережено результат встановлюється методом set_result(), зберігається шлях у полі resultpath. За допомогою методу import_classes() реалізовано імпорт назв об'єктів із текстового файлу. Зберігаються дані у полі classes. Методами add_class() реалізовано ручне додання назви об'єкту, що введено до спеціального поля. Метод delete_class() дозволяє вилучити назву об'єкту із списку, в якому обирається необхідний об'єкт. Метод detect_file() виконує розпізнавання об'єктів на зображенні. Методом save_label_image() реалізовано збереження та відображення зображення із об'єктами.

Методом класу MessageWindow реалізовано відображення інформаційних повідомлень у модулі. Клас MessageWindow успадковує клас UI_MessageWindow – таким чином реалізовано об'єднання графічної та програмної частини функціоналу виведення повідомлень. Клас не містить полів та методів, але при його ініціалізації потрібно вказати текст повідомлення.

					ІАЛЦ.045490.004 ПЗ	Арк.
						42
Змін.	Арк.	№ докум.	Підпис	Дата		

3.4. Забезпечення надійності програмного продукту.

Висока якість програмного забезпечення характеризується низькою відмовою виконання. Програма має бути протестована, точки аварійного завершення виконання мають бути виявлені. До аварійного завершення виконання програми призводять дії користувача, які можуть бути коректними, але є виключенням для програми. У розробленому модулі було оброблено велику кількість випадків завершення виконання програми. Користувачу надається інформування у вигляді повідомлень. Далі буде розглянено тексти повідомлень та випадки.

Надійність форми для розпізнавання об'єктів. забезпечується спеціальними перевірками у програмному коді. У випадку, якщо вхідні дані є некоректними програма не може продовжувати роботу. Про це програма повідомляє користувача інформаційними сповіщеннями.

На рисунку 3.11. представлено повідомлення яке виникає у випадку, якщо користувач не обрав теку для збереження файлу, або не відмітив галочку для відображення результату. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

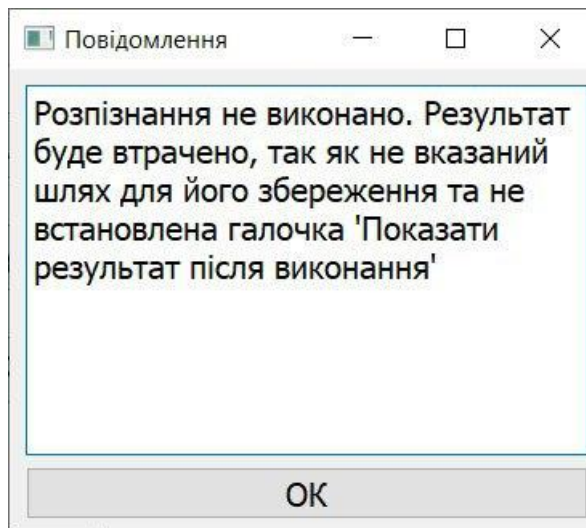


Рисунок 3.11 – Повідомлення про неможливість запуску розпізнання.

На рисунку 3.12 представлено повідомлення яке виникає у випадку, якщо користувач не обрав модель для розпізнання. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

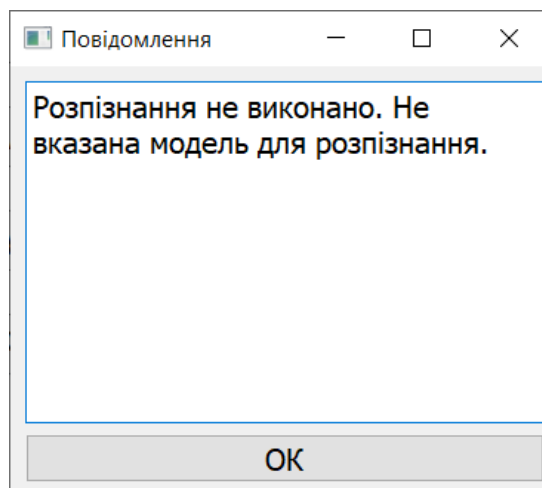


Рисунок 3.12 – Повідомлення про неможливість запуску розпізнання

На рисунку 3.13 представлено повідомлення яке виникає у випадку, якщо користувач не обрав зображення для розпізнання. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести

необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

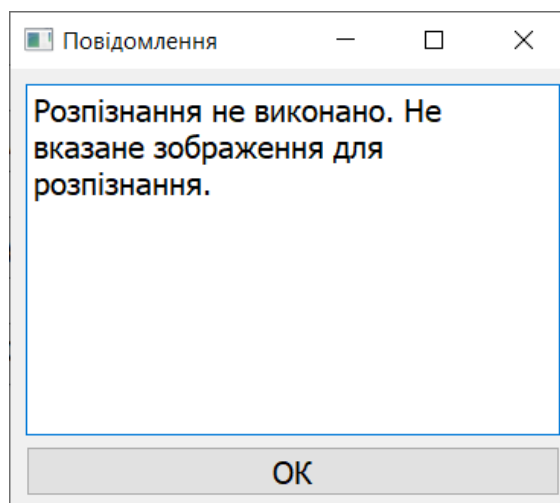


Рисунок 3.13 – Повідомлення про неможливість запуску розпізнання

На рисунку 3.14 представлено повідомлення яке виникає у випадку, якщо користувач не імпортував список об'єктів та не додавав їх власноруч. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

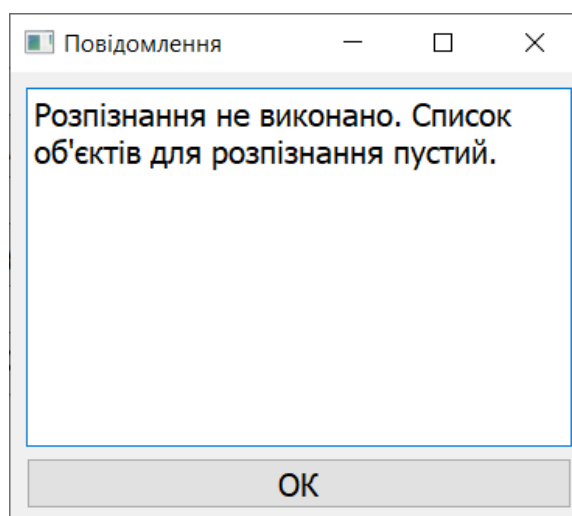


Рисунок 3.14 – Повідомлення про неможливість запуску розпізнання

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

45

На рисунку 3.15 представлено повідомлення, яке не є інформуванням про неможливість виконання програми, але є важливим попередженням. У випадку, якщо користувач обирає потужності власного пристрою для створення нової моделі нейронної мережі – користувач має бути певним, що потужність пристрою є прийнятною. Додатково, таке повідомлення інформує про те, що процес навчання потребує усіх ресурсів пристрою. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

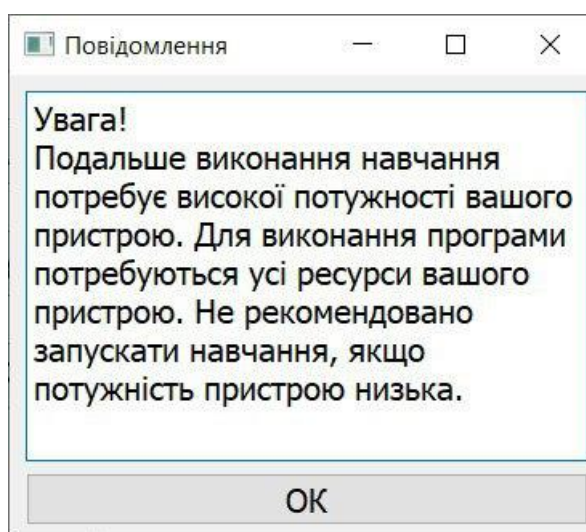


Рисунок 3.15 – Повідомлення, що попереджає про високу ресурсоємність навчання

На рисунку 3.16 представлено повідомлення, яке інформує про успішне завершення навчання нейронної моделі. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

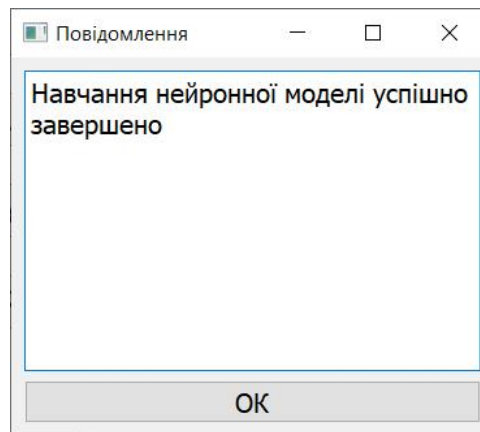


Рисунок 3.16 – Повідомлення про успішне навчання моделі

На рисунку 3.17 представлено повідомлення яке виникає у випадку, якщо користувач не обрав теку з даними для навчання. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

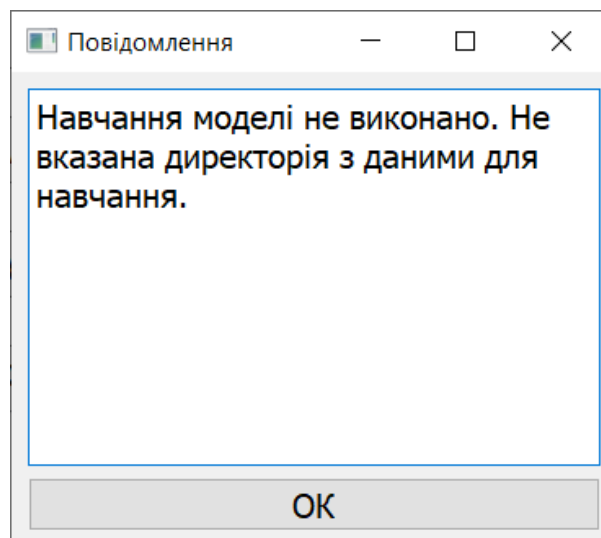


Рисунок 3.17 – Повідомлення про неможливість запуску навчання.

На рисунку 3.18 представлено повідомлення яке виникає у випадку, якщо користувач не обрав теку для збереження навченої моделі. Після закриття сповіщення користувачу відображається форма, в якій є можливість

внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

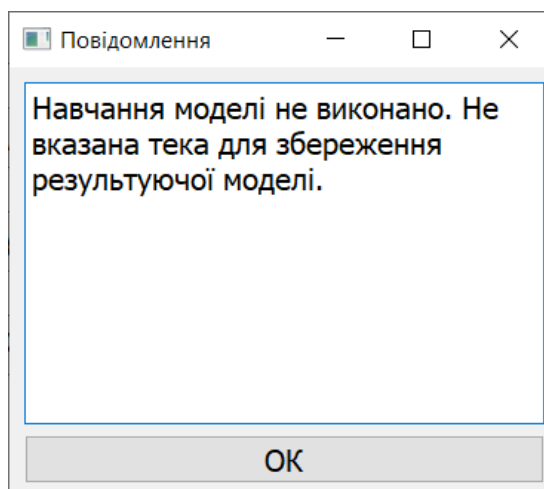


Рисунок 3.18 – Повідомлення про неможливість запуску навчання

На рисунку 3.19 представлено повідомлення яке виникає у випадку, якщо користувач не імпортував список об'єктів та не додавав їх власноруч. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

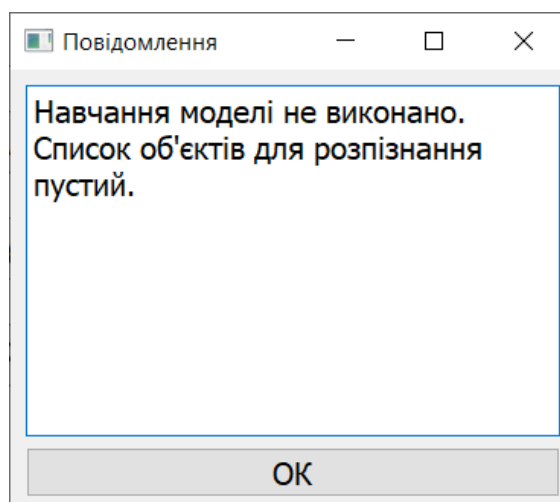


Рисунок 3.19 – Повідомлення про неможливість запуску навчання

На рисунку 3.20 представлено повідомлення яке виникає у випадку, якщо користувач не вибрав апаратні ресурси, які буде використано для навчання. Після закриття сповіщення користувачу відображається форма, в якій є можливість внести необхідні зміни. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

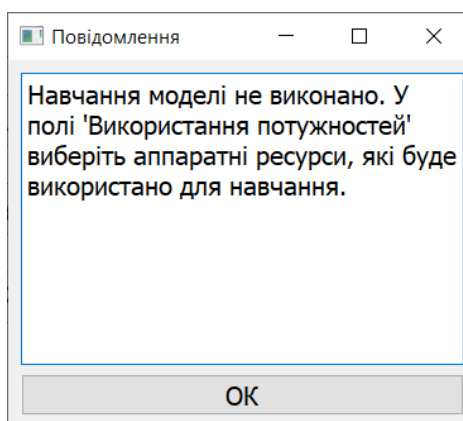


Рисунок 3.20 – Повідомлення про неможливість запуску навчання

На рисунку 3.21 представлено повідомлення, що виникає під час навчання моделі. Воно виникає у випадку, якщо користувач використовує вхідні дані з неправильним форматом представлення. Пропонується звернутися до довідки для допомоги. Закрити повідомлення можна кнопками «хрестик» - стандартна кнопка ОС Windows, або кнопкою «ОК».

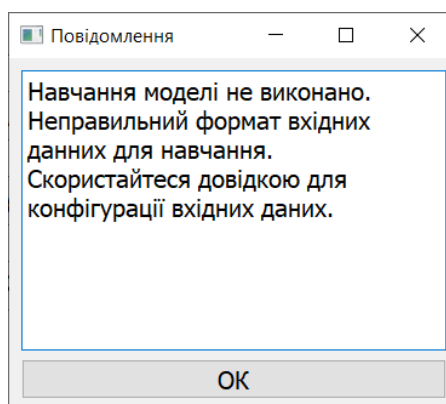


Рисунок 3.21 – Повідомлення про неправильні вхідні дані

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

49

ВИСНОВКИ ДО РОЗДІЛУ 3

В даному розділі описано розробку модуля розпізнавання зображень. На початку розробки сформовано архітектуру програми. По сформованій архітектурі розроблено класи, які були описані в діаграмі нотації UML. По сформованим класам описано розроблені функції та створено інтерфейс для програмного забезпечення. Для покращення якості програмного продукту забезпечено надійність виконання за допомогою системи відображення повідомлень та зупинки виконання програми.

					ІАЛЦ.045490.004 ПЗ	Арк.
						50
Змін.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

У дипломному проєкті розроблено модуль розпізнавання зображення УЗД.

Для розробки модуля було проаналізовано предметну область розпізнавання зображень. Проведено аналіз аномалій внутрішньоутробного розвитку, означено терміни скринінгу плоду та описано дефекти, що можуть бути знайдені під час УЗД скринінгу. Проведено аналіз медичних зображень, їх видів та способів отримання таких зображень.

Для постановки задачі до дипломного проєкту було проаналізовано аналог розроблюваного модуля, розглянено особливості та переваги цієї системи, виявлено недоліки та обмеження у розробки. На основі недоліків з метою їх вирішення було сформовано постановку задачі, які мають бути розв'язані у розроблюваному модулі.

Для вирішення постановлених завдань було розроблено модуль розпізнавання зображення УЗД: сформовано архітектуру програмного продукту, розроблено класи та представлено їх діаграму, розроблено інтерфейс та програмний код.

На даному етапі програмний продукт повністю готовий до використання у медичних закладах. Модуль для розпізнавання зображень УЗД можна вдосконалити новим функціоналом, таким як повна синхронізація даних із хмарними сховищами, розширення можливостей розпізнавання, пристосування програмного продукту до виявлення конкретних захворювань або вад розвитку.

					ІАЛЦ.045490.004 ПЗ	Арк.
						51
Змін.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ДЖЕРЕЛ

1. Підручник для студентів вищих фармацевтичних навчальних закладів і фармац. факультетів вищ. мед. навч. закладів. – Видання друге доп. та перероб. / А.І. Березнякова, та ін. Львів: 2010. – 530 с.
2. Патофізіологія: підручник (ВНЗ III—IV р. а.) / М.Н. Зайко, Ю.В. Биць, М.В. Кришталь та ін.; за ред. М.Н. Зайка, Ю.В. Биця, М.В. Кришталя. — 6-е вид., переробл. і допов. Київ 2017 – 736 с.
3. «Акустичн і медичні прилади»: Метод. Вказівки до викон. Курсового проекту та розрахунково-графічної роботи для студ. Напрямку підготовки 6.051003 - «Приладобудування» програми професійного спрямування «Медичні прилади і системи» / Уклад.: М.Ф.Терещенко, Г.С. Тимчик, І.О. Яковенко-К.: НТУУ «КПІ», 2016.-174с.
4. Педіатрія: підручник для студ. вищих мед. навч. Закладів П24 IV рівня акред. / за ред. Проф. О.В. Тяжкої, та ін. – Вид. 4-те, допрац. і допов. – Вінниця: Нова Книга, 2016 – 1152 с.
5. URL:
http://intranet.tdmu.edu.ua/data/kafedra/internal/informatika.../lectures_stud/uk/pharm/tpkz/ptn/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97%20%D1%83%20%D1%84%D0%B0%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97/2/05-%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D0%BC%D0%B5%D0%B4%D0%B8%D1%87%D0%BD%D0%B8%D1%85%20%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C.htm. (дата звернення 01.05.2020)

					ІАЛЦ.045490.004 ПЗ	Арк.
						52
Змін.	Арк.	№ докум.	Підпис	Дата		

6. INTRODUCTION TO MACHINE LEARNING / Nils J. Nilsson, etc.
Stanford, CA 94305, 1998 – 188 pp.
7. Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning
tools and techniques Morgan Kaufmann. Cambridge, MA 02139, 2011 - 664pp.
8. Masakazu Matsug, Katsuhiko Mori, Yusuke Mitari, Yuji Kaneda.
Subject independent facial expression recognition with robust facedetection using a
convolutional neural network. 2003.
9. URL: <https://www.intelligentultrasound.com/scannav-2/> (дата звернення
01.05.2020).
10. URL: <https://docs.python.org/2/index.html> (дата звернення
01.05.2020).
11. URL: R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object
Detection Algorithms — [Електронний ресурс]. — Режим доступу:
[https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-
algorithms-36d53571365e](https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e) (дата звернення 01.05.2020).
12. URL: <https://evergreens.com.ua/ua/articles/cnn.html> (дата звернення
01.05.2020).
13. URL: <https://detecto.readthedocs.io/en/latest/> (дата звернення
01.05.2020).
14. URL: <https://pytorch.org/tutorials/> (дата звернення 01.05.2020).
15. Qt Documentation Archive [Електронний ресурс] Режим доступу:
<http://doc.qt.digia.com/> (дата звернення 01.05.2020).
16. URL: <https://jupyter.org/about> (дата звернення 01.05.2020).